

## Estruturas de Dados e Algoritmos - Prova 01

Nome:

Matrícula:

Data:

---

### OBSERVAÇÕES:

- (A) A PROVA É INDIVIDUAL E SEM CONSULTA, SENDO VEDADO O USO DE TELEFONES CELULARES OU MATERIAIS DE APOIO.
  - (B) A INTERPRETAÇÃO DOS COMANDOS DOS PROBLEMAS FAZ PARTE DA AVALIAÇÃO.
  - (C) A SOLUÇÃO DE CADA PROBLEMA DEVE ESTAR EM UM ARQUIVO CUJO NOME DEVE SER: {NÚMERO DO PROBLEMA}-{MATRÍCULA SEM BARRA}.C (OU .CPP).
  - (D) AS ENTRADAS DO PROBLEMA DEVEM SER LIDAS DA ENTRADA **PADRÃO** DO SISTEMA (CONSOLE), E ESCRITAS NA SAÍDA **PADRÃO**.
  - (E) PARA VARIÁVEIS DO TIPO PONTO FLUANTE, UTILIZAR O TIPO PRIMITIVO **DOUBLE** PARA EVITAR DIFERENÇAS DE PRECISÃO.
- 

1. Um número natural é denominado um *número de Thabit*  $T_k$  se ele pode ser escrito na forma

$$T_k = 3 \cdot 2^k - 1,$$

onde  $k$  é um número inteiro não-negativo. Os primeiros quatro números de Thabit são 2, 5, 11, 23.

Dado um número natural  $n$ , determine se ele é ou não um número de Thabit.

**Variáveis:**

Símbolo	Tipo	Descrição	Valores válidos
$n$	<b>int</b>	Número natural a ser identificado como de Thabit ou não	$1 \leq n \leq 1.000.000$
$c$	<b>char</b>	Classificação do número: $T$ para número de Thabit e $N$ caso não seja número de Thabit	$c \in \{N, T\}$

**Entrada:** o programa deverá receber a seguinte entrada:

$n$

**Saída:** a saída do programa deverá ser o valor de  $c$ , seguido de uma quebra de linha.

**Exemplos de entradas e saídas esperadas:**

Entradas	Saídas
1	N
2	T
1533	N
3071	T

2. Denomina-se **palíndromo** uma palavra, frase, número ou qualquer sequência de símbolos que possam ser lidos tanto da esquerda para a direita quanto da direita para a esquerda. Em geral, não são levados em consideração espaços, pontuações e acentuações.

Como exemplos de palíndromos, temos a palavra REVIVER, o número 12321 e a frase “*Socorram-me, subi no ônibus em Marrocos*”.

Dada uma palavra  $P$ , determine se ele é ou não um palíndromo.

**Entrada:** A entrada consiste em uma linha com o número  $N$  e a palavra  $P$ , separados por um espaço e seguidos de uma quebra de linha, onde  $N$  é o número de caracteres que compõem a palavra. Pode-se considerar que  $P$  não conterá espaços, pontuações ou acentuações, e que  $N$  é um inteiro positivo não nulo.

**Saída:** A saída do programa deverá ser a mensagem “ $P$  E UM PALINDROMO” ou “ $P$  NAO E UM PALINDROMO”, de acordo com  $P$ , seguida de uma quebra de linha.

Entradas	Saídas
6 MUSSUM	MUSSUM E UM PALINDROMO
4 SAIA	SAIA NAO E UM PALINDROMO
5 SAIAS	SAIAS E UM PALINDROMO
8 SOPAPO	SOPAPO NAO E UM PALINDROMO

3. Considere que uma impressora esteja configurada para realizar impressão em **dupla face** (isto é, dos dois lados de uma folha). Uma vez definido o documento a ser impresso e o intervalo de páginas a serem impressas (isto é, a página **inicial** e o **total** de páginas a serem impressas), a impressora procede da seguinte forma: ela imprime a primeira página do intervalo no **verso** da página, e a página seguinte na **frente**. A folha impressa é então liberada na bandeja para recolhimento, ficando o verso para **baixo** e a frente para **cima**. O processo continua deste modo até que todas as páginas solicitadas tenham sido impressas.

Considerando que nenhuma folha será removida da bandeja antes do término da impressão, determine o número da página impressa na face que estará visível no **topo** do monte formado na bandeja. Caso a face esteja em branco, considere que o número da página é 0 (zero).

**Variáveis:**

Símbolo	Tipo	Descrição	Valores válidos
$p$	<b>int</b>	Número da primeira página a ser impressa	$1 \leq p \leq 10.000$
$q$	<b>int</b>	Total de páginas a serem impressas	$1 \leq q \leq 10.000$
$n$	<b>int</b>	Número da página impressa na face visível do monte, ao final da impressão	$0 \leq n < 20.000$

**Entrada:** o programa deverá receber a seguinte entrada:

$p \ q$

**Saída:** a saída do programa deverá ser o valor de  $n$ , seguido de uma quebra de linha.

**Exemplos de entradas e saídas esperadas:**

Entradas	Saídas
1 4	4
2 6	7
3 1	0
8 5	0

4. Em um sistema distribuído, os processos possuem um identificador numérico. No processo de inicialização do sistema, são carregados  $M$  processos, que são armazenados em um vetor, e cada processo é delegado a um dos terminais de processamento.

A medida que cada terminal termina o processo que foi a ele atribuído, ele envia uma mensagem ao sistema, que contém o identificador do processo terminado e o caractere  $t$ . Ao longo do processamento, um terminal pode exigir a criação de novos processos a serem adicionados ao vetor de processos. Neste caso, a mensagem conterá um novo identificador, seguido do caractere  $n$ .

As operações de remoção e inserção de processos no vetor do sistema seguem as seguintes regras:

- A inserção é feita sempre ao final do vetor, após o último processo pendente.
- A remoção é feita através da troca dos identificadores do processo a ser removido com do processo que ocupa a última posição válida do vetor. Após a permuta, o último elemento é removido.
- O vetor do sistema tem uma capacidade máxima de  $N$  processos pendentes. Caso uma inserção tente ser realizada com o vetor completamente cheio, o processo a ser inserido deve ser descartado.
- Também deverá ser ignorada uma tentativa de remoção quando o vetor de processos pendentes estiver vazio ou o identificador do processo fornecido não estiver presente no vetor.

Neste contexto, dada a capacidade máxima do vetor, o seu estado inicial (do elemento da posição 0 ao elemento da posição  $M - 1$ ), a quantidade de requisições de inserção e remoção e a descrição de cada requisição, determine o estado final do vetor.

**Entrada:** A entrada consiste em uma única linha com a capacidade máxima  $N$  do vetor, o número  $M$  de processos a serem carregados na inicialização do sistema, seguidos pelos seus respectivos identificadores, o número  $R$  de requisições de inserção e remoção e as descrições das mensagens de inserção e remoção. Todos os valores são separados por espaços em branco,  $N$ ,  $M$  e  $R$  são inteiros positivos e a entrada termina com uma quebra de linhas.

**Saída:** A saída do programa deverá conter a mensagem “ $V = [id_1, id_2, \dots, id_T]$ ”, onde  $id_i$  é o identificador do  $i$ -ésimo processo no vetor. Cada identificador deve ser seguido de uma vírgula e um espaço em branco, exceto o último, e todos eles devem estar entre colchetes. Ao final da mensagem deve ser impressa uma quebra de linhas.

Entradas	Saídas
10 2 1 2 1 3 n	$V = [1, 2, 3]$
10 3 3 1 2 3 3 t 4 n 5 n	$V = [2, 1, 4, 5]$
4 3 5 1 8 4 2 n 4 n 6 n 1 t	$V = [5, 2, 8]$
4 2 5 3 3 5 t 3 t 1 t	$V = []$

**Bom estudo!**